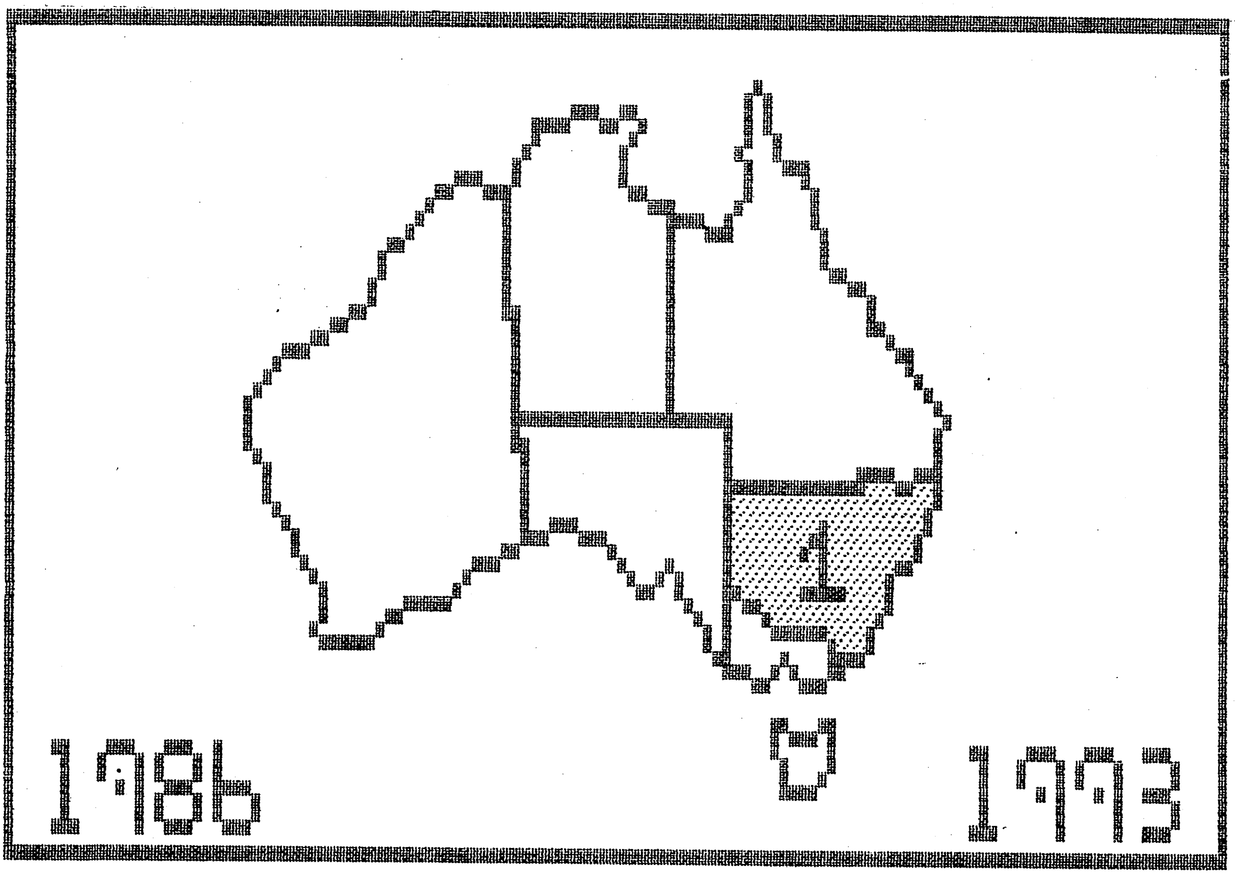
JUNE 1993 ISSUE # 43

HUNTER VALLEY VZ JOURNAL

HAPPY 7TH BIRTHDAY



PRODUCED BY JOSEPH P. LEON

A NON PROFIT PUBLICATION

FRONT COVER

MY THANKS TO JASON OAKLEY FOR DESIGNING TOP HALF WHILE THE REST IS SELF EXPLANATORY. FOR THOSE WHO CAN'T WORK OUT WHAT THE 1 STANDS FOR THEN TURN TO PAGE 3.

HELP - SELL & TELL — PAGE 3

7TH BIRTHDAY, APOLOGIES, NEXT/FUTURE AND BACK ISSUES.

BUGS REPORT FOR DOS V1.2 BY LESLIE MILBURN — PAGE 4

LESLIE HAS FOUND SOME BUGS IN DOS AND IS SHARING IT WITH OTHER DOS USERS. THANKS LESLIE.

DISABLING BREAK KEY BY LESLIE MILBURN — PAGES 4-8

LESLIE SHOWS US HOW TO CREATE BREAK KEY PROOF BASIC PROGRAMS. IF YOU HAVE WANTED TO LOCK UP YOUR PROGRAMS FOR WHATEVER REASON YOU CAN NOW DO SO.

INTRODUCTION TO PROGRAMMING PART II BY BOB KITCH — PAGES 8-11

BOB CONCLUDES HIS ESSAY INTO MORE STRUCTURED PROGRAMMING AND MOST OF US SHOULD BENEFIT FROM HIS EFFORTS.

WP FILE RETRIEVE BY DAVE MITCHELL — PAGES 11-15

THE ABOVE DISK UTILITY HAS BEEN DESIGNED TO RESURRECT WP FILE DISKS WHOSE TRACK ZERO ARE UNREADABLE. I FOUND OUT IT CAN DO MORE THAN THAT.

PREVENTITIVE DISK MAINTAINANCE — PAGE 15

DAVE MITCHELL SOFTWARE FOR SALE — PAGE 16

E&F DISK WP PATCH3.3
DISK FILER CATALOGUER
EXTENDED DOS VERSION 1.3
MENU/FILE COPIER

USER GROUPS - CONTRIBUTIONS SUBSCRIPTIONS — PAGE 16

HUNTER VALLEY VZ JOURNAL INDEX PART I - SUPPLEMENT — PAGES 01-05

THE LAST INDEX COVERING ISSUES 1 TO 24 WAS PUBLISHED IN ISSUE 24, MAY/JUNE 1989. THIS CURRENT INDEX COVERS ISSUES 1 TO 20 WITH 21 TO 44 PLANNED FOR NEXT ISSUE. THE REASON FOR REPEATING THE FORMER IS THAT THE MAJORITY OF REMAINING SUBSCRIBERS WERE NOT MEMBERS AT THAT TIME.

DISCLAIMER: EVERY EFFORT IS MADE TO INSURE THE ACCURACY OF INFORMATION CONTAINED WITHIN BE IT GENERAL, TECHNICAL, PROGRAMMING, ETC. NO RESPONSIBILITY CAN BE ACCEPTED BY HUNTER VALLEY VZ USERS' GROUP OR AUTHOR AS A RESULT OF APPLYING SUCH INFORMATION IN PRACTICE.

FRONT COVER . . .

FIRST, WITH THIS ISSUE WE CELEBRATE 7 YEARS OF OPERATION OF THE HUNTER VALLEY VZ USER'S GROUP AND THE HUNTER VALLEY VZ JOURNAL OF WHICH I HAVE BEEN EDITOR FOR THE LAST 6.5 YEARS.

MY **SINCERE THANKS** TO ALL SUPPORTERS OF THE JOURNAL, BOTH CONTRIBUTORS AND SUBSCRIBERS FOR THE JOURNAL COULD NOT HAVE SURVIVED SO LONG WITHOUT YOU. **THANK YOU ALL.**

SECOND, THE NO 1 IN THE MAP OF AUSTRALIA SIGNIFIES:

1) THE FIRST STATE.
2) THE NO 1 VZ PUBLICATION SINCE ITS INCEPTION.
3) THE ONLY VZ PUBLICATION LEFT IN AUSTRALIA.

7 YEARS AND STILL GOING STRONG

LONG LIVE THE VZ

APOLOGIES . . .

ONE OF THE REASONS FOR THE LATENESS OF THIS ISSUE IS THAT THE PHOTOSTAT MACHINE HAS BEEN BROKEN DOWN FOR OVER TWO MONTHS AND THEY ENDED UP GETTING A NEW A FEW DAYS AGO. SECOND AS SOME OF YOU WERE AWARE I HAD A HOUSE BUILT TO LOCK UP STAGE AND AM FINISHING IT MYSELF WITH THE HELP OF FAMILY AND FRIENDS.

THIS HAS AFFECTED MY HEALTH IN VARIOUS WAYS WITH THE WORST BEING INCREASED PAIN AND EXHAUSTION TAKING OVER AFTER A DAYS WORK. I HOPE YOU'LL BEAR WITH ME TILL THE HOUSE IS FINISHED WHICH SHOULD BE LATE AUGUST. HAVE STARTED FINAL PLASTERING WITH PAINTING TO FOLLOW.

AT LONG LAST I'VE FOUND OUT THE NAME OF MY MEDICAL CONDITION. IT IS POST TRAUMA CHRONIC PAIN SYNDROME AFFECTING ALL JOINTS AND MOST MUSCLES. ONCE I MOVE TO MY NEW HOUSE I'LL BE ABLE TO RELAX AND RECUPERATE AT MY LEISURE AND CATCH UP WITH WITH VZ JOURNAL AND GET BACK INTO ELECTRONIC PROJECTS AGAIN.

NEXT/FUTURE ISSUE/S . . .

EXPANDED PUBLIC DOMAIN LISTING WITH NEW AUTHORS GOING PUBLIC.
BASIC DOS UTILITY FOR BASIC PROGRAMMERS.
HOW TO USE DOS M/C ROUTINES FROM AND IN BASIC.
MULTIPLE USR CALLS.
HI-RES 2K/64K GRAPHICS EDITOR.
JOURNAL INDEX PART II

BACK ISSUES . . .

MOST BACK ISSUES ARE IN STOCK AND AVAILABLE AT FOLLOWING PRICES.

CURRENT ISSUE - $3.50 EACH - INCLUDES POSTAGE
1 - 5 ISSUES - $3.00 EACH - INCLUDES POSTAGE
6 OR MORE ISSUES - $2.75 EACH - INCLUDES POSTAGE

DOS BUGS REPORT FOR DOS V1.2

BY LESLIE MILBURN

1. IF YOU USE CLOSE IN IMMEDIATE MODE, THE FILE BUFFER IS NOT FLUSHED TO DISK.

2. WHEN SAVING STRINGS TO DISK USING PR#, THE STRING MUST NOT CONTAIN A COMMA, COLON OR CARRIAGE RETURN.

3. THE 116 BYTES OF "UNUSED MEMORY" IN THE COMMUNICATIONS REGION (7A29H - 7A9CH) IS USED BY THE IN# COMMAND. THIS MEANS THAT KSCAN, MOUSE DRIVER 1.04 AND FIND WILL BE DESTROYED IF A BASIC PROGRAM USING IN# IS RUN.

4. THE MAXIMUM NUMBER OF CHARACTERS (BYTES) THAT CAN BE READ BY ONE IN# COMMAND IS 199. THIS IS REGARDLESS OF THE NUMBER OF VARIABLE ARGUEMENTS.

5. DO NOT USE ANY OTHER DISK COMMANDS WHILE A DATA FILE IS OPEN, OTHER THAN OPEN, CLOSE, IN# AND PR#, OTHERWISE CORRUPT DATA COULD RESULT.

6. WHEN A BASIC PROGRAM PERFORMS A LOT OF STRING MANIPULATION, THE "GARBAGE COLLECTOR" OCCASIONALLY TAKES CONTROL AND REARRANGES THE BASIC STRING SPACE. IF THIS HAPPENS WHILE YOU ARE SAVING STRINGS TO DISK VIA PR#, A CORRUPT DATA FILE WILL OCCUR AS THE STRINGS HAVE BEEN MOVED WITHOUT PR# KNOWING ABOUT IT.

IF ANYBODY HAS FOUND ANY OTHER BUGS IN ANY DOS SYSTEM, PLEASE INFORM THE EDITOR WHO CAN PASS THE INFORMATION ON TO OTHER USERS.

CREATING BREAK KEY PROOF BASIC PROGRAMS BY LESLIE MILBURN

WHEN I FIRST BOUGHT MY VZ NEARLY 9 YEARS AGO, ONE OF THE FIRST THINGS I WANTED TO DO WAS TO DISABLE THE BREAK KEY TO STOP PEOPLE LISTING MY BASIC PROGRAMS. WHEN I ENQUIRED FURTHER I WAS TOLD THAT IT COULD NOT BE DONE. HOWEVER, IT CAN AND FAIRLY EASILY!

ABOUT THE BREAK KEY

THE MAIN PROBLEM WHEN DEALING WITH THE BREAK KEY IS THAT MANY OF THE ROM ROUTINES CHECK FOR IT IN DIFFERENT WAYS.

THE ORIGINAL LEVEL II BASIC ROM ROUTINES CHECK THE BREAK KEY VIA TWO MAIN ROUTINES AND REPLACING THESE IS NOT DIFFICULT. KSCAN (REFER JOURNAL #38) DOES EXACTLY THAT.

UNFORTUNATELY, THE ROM ROUTINES WHICH HAVE BEEN ADDED SPECIFICALLY FOR THE VZ (I.E. EXTENSIONS TO THE LEVEL II BASIC) HAVE THE BREAK KEY CHECKS IN ROM AND CANNOT BE INTERCEPTED. THESE ROUTINES CORRESPOND TO THE FOLLOWING BASIC COMMANDS:-

CLOAD, COPY, CRUN, CSAVE, INPUT#, PRINT#, LPRINT, SOUND, VERIFY

NOTICE THAT ALL OF THESE COMMANDS DEAL WITH EITHER CASSETTE, PRINTER OR SPEAKER I/O. ALSO NOTE THAT IF YOU HAVE A DISK DRIVE ATTACHED, ALL DISK BASIC COMMANDS CHECK THAT THE MINUS KEY IS PRESSED (NOT CTRL-MINUS). THIS MAY NOT BE THE CASE FOR YOUR DISK CONTROLLER AS VARIATIONS EXIST.

DISABLING THE BREAK KEY

IF YOU HAVE KSCAN INSTALLED THEN THE BREAK KEY CAN BE DISABLED BY STORING A NULL IN THE KEY TABLE FOR THE CTRL-MINUS COMBINATION USING SETKEY(2,2,0,2).

FOR THOSE OF YOU WHO DO NOT HAVE (AND DON'T WISH TO HAVE) KSCAN INSTALLED, INCLUDED IS A SIMPLE RELOCATABLE ROUTINE TO DISABLE THE BREAK KEY. NOTE THAT ANY INSTALLED INTERRUPT ROUTINES ARE ALSO DISABLED.

CREATING AN AUTO-EXECUTE BASIC PROGRAM

TO MAKE A BASIC PROGRAM AUTOMATICALLY EXECUTE FROM DISK IS QUITE SIMPLE. THE METHOD IS TO CREATE A MACHINE CODE HEADER AND SAVE IT AND THE BASIC PROGRAM TO DISK AS A BINARY FILE. TO RUN THE PROGRAM BRUN IS USED INSTEAD OF RUN AND THE M/C HEADER IS EXECUTED.

FOR EXAMPLE:-

BASIC PROGRAM: 10 PRINT"HELLO"

```
START ADDRESS = 7AE9H (31465)
END ADDRESS   = 7AF8H (31480)
```

```
M/C HEADER:  LD HL,7AE9H  ;BASIC PROGRAM START ADDRESS.
    LD (78A4H),HL ;STORE IN START OF BASIC POINTER.
    EX DE,HL
    JP 36E9H  ;EXECUTE BASIC PROGRAM.
```

TO MAKE THE EXAMPLE BASIC PROGRAM AUTOMATICALLY RUN, THE PROCEDURE IS AS FOLLOWS:-

(1) POKE 31376,33:POKE 31377,233:POKE 31378,122
POKE 31379,34:POKE 31380,164:POKE 31381,120
POKE 31382,235
POKE 31383,195:POKE 31384,233:POKE 31385,54.

THIS STORES THE HEADER IN AN UNUSED AREA OF COMMUNICATIONS RAM AT ADDRESS 7A90H

(2) BSAVE"PROGRAM",7A90,7AF8H (END ADDRESS OF BASIC PROGRAM)

THIS SAVES THE HEADER & BASIC PROGRAM TO DISK AS A BINARY FILE.

(3) TO RUN THE PROGRAM SIMPLY TYPE BRUN"PROGRAM"

CREATING A BREAK KEY PROOF BASIC PROGRAM

TO CREATE A BREAK KEY PROOF PROGRAM IS NOW EASY. OUR GOAL IS TO DISABLE THE BREAK KEY AS SOON AS THE PROGRAM IS LOADED BUT BEFORE THE BASIC INTERPRETER IS CALLED. THIS CAN BE ACHIEVED BY CREATING A SLIGHTLY DIFFERENT M/C ROUTINE. IT SHOULD BE LIKE THE FOLLOWING:-

```
CALL INIT ;INSTALL THE DISABLE BREAK KEY ROUTINE
LD HL, BASIC PROG START ADDR.
LD (78A4H),HL
EX DE,HL
JP 36E9H ;EXECUTE THE BASIC PROGRAM
```

AS YOU CAN SEE, THIS IS VERY SIMILAR TO THE PREVIOUS M/C ROUTINE EXCEPT THAT THE BREAK KEY ROUTINE IS INSTALLED PRIOR TO RUNNING THE BASIC PROGRAM. THIS MEANS THAT THE BREAK KEY ROUTINE MUST BE SAVED ALONG WITH THE HEADER & BASIC PROGRAM. SEE BELOW FOR FURTHER DETAILS.

USING THIS TECHNIQUE WE CAN NOW SAVE A BASIC PROGRAM AS AN AUTO-EXECUTING PROGRAM WHICH DISABLES THE BREAK KEY. IMMEDIATE MODE CAN ONLY BE ENTERED IN THREE CASES:-

(1) THE BREAK KEY WAS PRESSED WHILE THE PROGRAM WAS PERFORMING CASSETTE, PRINTER, SPEAKER OR DISK I/O

(2) A COMMAND WAS EXECUTED WHICH RETURNS TO IMMEDIATE MODE. THESE COMMANDS ARE:-

LIST, LLIST, STOP, END & NEW

(3) AN ERROR OCCURRED AND NO ERROR HANDLER WAS INSTALLED.

(I.E. NO ON ERROR COMMAND WAS INCLUDED IN THE PROGRAM).

THEREFORE, YOUR PROGRAM CANNOT BE LISTED ONCE RUN UNLESS ONE OF THE ABOVE CONDITIONS OCCUR.

WHAT IF THE PROGRAM IS BLOAD'ED NOT BRUN?

AS MENTIONED ABOVE, THE BREAK KEY ROUTINE MUST BE SAVED ALONG WITH THE BASIC PROGRAM AND M/C HEADER. THE TECHNICAL REFERENCE MANUAL DISCUSSES THREE METHODS FOR RESERVING MEMORY FOR A MACHINE CODE SUBROUTINE. TWO OF THESE ARE SUITABLE FOR OUR PURPOSE.

ONE METHOD INVOLVES INCREASING THE END OF BASIC POINTER AND SAVING THE M/C ROUTINE IN THE GAP CREATED AFTER THE BASIC PROGRAM. ONE DISADVANTAGE OF THIS METHOD IS THAT AS THE BASIC PROGRAM GROWS, THE M/C START ADDRESS CHANGES.

THE OTHER METHOD INVOLVES MOVING THE START OF BASIC POINTER THUS RESERVING MEMORY BETWEEN THE END OF THE COMMUNICATIONS REGION AND THE NEW START OF BASIC.

THE LATTER METHOD IS MORE FAVOURABLE FOR AN IMPORTANT REASON, WHAT IF THE BREAK PROTECTED PROGRAM IS BLOAD'ED NOT BRUN. IF THIS OCCURS THE PROGRAM CAN BE LISTED BUT NOT ALTERED AS THE START AND END OF BASIC POINTERS ARE NOT CORRECT. TO PREVENT THE PROGRAM FROM BEING ABLE TO BE LISTED WE MUST FOOL THE BASIC INTERPRETER INTO BELIEVING THAT THE PROGRAM AREA IS EMPTY.

THIS IS DONE BY SETTING THE FIRST TWO BYTES IN THE PROGRAM AREA TO NULL BYTES. AS MOST USERS DO NOT BOTHER TO ALTER THE START OF BASIC POINTER FROM ITS DEFAULT VALUE WE CAN ASSUME THAT IT IS SET TO 31465 AND STORE NULL IN 31465 & 31466.

WE CAN NOW LIST THE STEPS TO CREATE A "BREAK KEY PROOF, AUTO-STARTING, NON-LISTABLE" BASIC PROGRAM. THESE ARE AS FOLLOWS:-

(1) INCREASE THE START OF BASIC POINTER AT 30884/5 SUFFICIENTLY TO STORE THE M/C ROUTINES.

(2) TYPE IN THE BASIC PROGRAM. NOTE THAT THE PROGRAM CANNOT BE LOADED FROM TAPE OR DISK WITHOUT THE AID OF A UTILITY. THIS IS BECAUSE PROGRAMS ARE ALWAYS LOADED BACK TO THE SAME ADDRESS THAT THEY WERE SAVED FROM.

(3) POKE 31465,0:POKE 31466,0 TO FOOL THE BASIC INTERPRETER INTO BELIEVING THE BASIC PROGRAM AREA IS EMPTY.

(4) STORE THE M/C HEADER ROUTINE, WHICH DISABLES THE BREAK KEY, AT ADDRESS 31467 (7AEBH).

(5) PRINT PEEK(30969)+256*PEEK(30970) TO GET THE END OF BASIC PROGRAM ADDRESS AND CONVERT IT TO HEX.

(6) BSAVE"PROGRAM",7AE9,END OF BASIC PROGRAM ADDRESS.

IN CONCLUSION:-

THIS ARTICLE SUGGESTS A VERY SIMPLE METHOD OF PROGRAM PROTECTION. IT DOES NOT ADDRESS THE PROBLEM OF DISK COPYING OR THE USE OF DISK EDITOR PROGRAMS.

THE TECHNIQUE OF PROVIDING MACHINE CODE HEADER ROUTINES COULD BE USED TO ASK FOR A PASSWORD BEFORE EXECUTING THE BASIC PROGRAM OR CHECKING IF A REQUIRED UTILITY IS LOADED SUCH AS KSCAN OR A MOUSE DRIVER.

BREAK-KEY DISABLE SOURCE CODE

```
;*****************************
; BREAK-KEY DISABLE
; UTILITY VERSION 2.0
;*****************************
;
;/ORG=7AE9H
;
;*****************************
;CONSTANT DEFINITIONS
;
TOM  EQU  78B1H
SSPC EQU  78A0H
INTX EQU  30845
SBAS EQU  78A4H
NEW  EQU  1B4DH
BAS  EQU  1A19H
ORG  EQU  7AE9H
KDCB EQU  7816H
;
;*****************************
;INIT: THIS IS THE MAIN
;      ENTRY POINT INTO
;      THIS UTILITY.
;
INIT DI
     LD   SP,7700H
;LOWER TOM AND COPY ROUTINE
     LD   HL,(TOM)
     LD   BC,END-INTR
;CLEAR CARRY FLAG
     OR   A
     SBC  HL,BC
     LD   (TOM),HL
;SAVE NEW TOM
     PUSH HL
;SET STRING SPACE TO 50
;BYTES
     LD   DE,50
     OR   A
     SBC  HL,DE
     LD   (SSPC),HL
;RESTORE TOM POINTER
     POP  DE
     INC  DE
;SAVE POINTER TO ROUTINE
     PUSH DE
     LD   HL,INTR
;COPY ROUTINE INTO RESERVED
;AREA.
     LDIR
;SET INTERRUPT EXIT TO POINT
;TO THE NEW INTERRUPT
;ROUTINE.
     LD   A,195
     LD   (INTX),A
;RESTORE POINTER TO ROUTINE.
     POP  HL
     LD   (INTX+1),HL
;SET KEY SCAN ROUTINE PTR.
     LD   DE,SCAN-INTR
     ADD  HL,DE
     LD   (KDCB),HL
;RESET START OF BASIC PTR
     LD   HL,ORG
     LD   (SBAS),HL
;EXIT TO BASIC
     CALL NEW
     EI
     JP   BAS
;*****************************
```

BREAK-KEY SOURCE CODE CONT.


```
;INTR: NEW INTERRUPT
;       ROUTINE.
;
;       NOTE: REGISTERS HAVE
;             BEEN SAVED UPON
;             ENTRY TO THIS
;             FUNCTION.
;
INTR CALL 3F7BH
;DISPLAY THE CURSOR
     CALL 2EDCH
;SCAN THE KEYBOARD
     CALL 002BH
;SAVE THE KEY CODE
     PUSH AF
;IF IN INPUT PHASE THEN
;DISPLAY CHARACTER, UPDATE
;SCREEN POSITION, ETC.
     LD   HL,7839H
     BIT  0,(HL)
     CALL Z,301BH
;RESTORE KEY CODE.
     POP  AF
;SOUND A BEEP IF KEY WAS
;PRESSED.
     CALL 3430H
;REMOVE RETURN ADDRESS SO
;THAT THE ROM ROUTINE IS
;BY-PASSED.
     POP  HL
;RESTORE PREVIOUSLY SAVED
;REGISTERS.
     POP  HL
     POP  DE
     POP  BC
     POP  AF
;ENABLE INT'S AND RETURN.
     EI
     RETI
;****************************
;SCAN: THIS FUNCTION SCANS
;      THE KEYBOARD.
;
SCAN CALL 2EFDH
     CP   1
     RET  NZ
     LD   A,0
     RET
END  EQU  $
```

INTRODUCTION TO PROGRAMMING PART II BY BOB KITCH

AS MENTIONED IN PART 1 OF THIS SERIES, THE PROGRAMMING TASK IS A LARGE AND COMPLEX FEAT OF ORGANIZATION AND REQUIRES A WIDE RANGE OF SKILLS. IT IS POSSIBLE, AND BEST, TO BREAK THE TASK DOWN INTO SIX SEGMENTS - EACH OF WHICH MUST BE THOUGHT ABOUT, PLANNED AND THEN CARRIED OUT TO ENSURE THE SUCCESSFUL COMPLETION OF A SOFTWARE PROJECT.

EVEN A SMALL PROGRAM REQUIRES THAT A CURSORY CONSIDERATION OF THE SIX SEGMENTS BE MADE - ALTHOUGH SOME OF THEM MAY BE QUICKLY PASSED OVER AS TRIVIAL. BUT IT IS CERTAIN THAT LARGER PROGRAMS (MORE THAN 200 LINES) REQUIRE CAREFUL PLANNING FOR SUCCESS.

BEFORE DESCRIBING THE SIX STEPS, IT IS WORTH THINKING ABOUT "WHAT MAKES A GOOD PROGRAM?"

A PROGRAM MAY BE JUDGED FROM A NUMBER OF DIFFERENT STANDPOINTS; EACH IS NOT NECESSARILY MUTUALLY EXCLUSIVE AND SOMETIMES SOME CONFLICTS REQUIRE THAT A TRADE-OFF BE MADE.

THE FIRST CRITERIA IS THAT A PROGRAM SHOULD BE EFFICIENT. EFFICIENCY CAN BE CONSIDERED FROM A NUMBER OF VARYING VIEW POINTS. FOR EXAMPLE, OPTIMIZATION OF THE RUN-TIME CAN BE CONSIDERED AS EFFICIENT. ALSO, REDUCTION IN STORAGE REQUIREMENTS FOR BOTH PROGRAM CODE AND VARIABLES CAN BE CONSIDERED AS EFFICIENT PROGRAMMING.

FURTHERMORE, AND PARTICULARLY IF ONE IS DEVELOPING SOFTWARE COMMERCIALLY, THEN EFFICIENCY CAN BE MEASURED IN TERMS OF THE ACTUAL TIME REQUIRED TO GET AN APPLICATIONS PROGRAM RUNNING AND THE EASE OF MAINTENANCE OF THAT CODE. THE USE OF APPROPRIATE DATA TYPES AND DATA STRUCTURES CAN GREATLY IMPROVE THE EFFICIENCY OF A PROGRAM. THE SELECTION OF A SUITABLE ALGORITHM CAN ALSO ASSIST. FINALLY, EASE OF DEBUGGING SO THAT THE PROGRAM CAN BE UPDATED OR MODIFIED MAY BE CONSIDERED DESIRABLE.

THE SECOND CRITERIA IS GENERALITY AND IT IS HERE PERHAPS THAT SO MANY PROGRAMS "SCORE" SO POORLY. RATHER THAN A PROGRAM BEING WRITTEN TO SOLVE A PARTICULAR CHORE, IT SHOULD BE BROADLY WRITTEN TO HANDLE A WIDE RANGE OF PROBLEMS. THE USE OF SUBROUTINES AND FUNCTIONS DEVELOPED AND DEBUGGED PREVIOUSLY CAN ENORMOUSLY IMPROVE PROGRAMMING PRODUCTIVITY. OFTEN A SIMPLE SUSTITUTION OF A VARIABLE FOR A CONSTANT IN A PROGRAM CAN BROADEN THE THE APPLICABILITY OF THE PROGRAM SIGNIFICANTLY.

THE FINAL CRITERIA IS ELEGANCE, WHICH IS A LITTLE HARDER TO BOTH DEFINE AND ACHIEVE. AN ELEGANT PROGRAM IS ONE THAT IS SIMPLE AND INGENIOUS, AND POSSIBLY USES AN ALGORITHM OR DATA STRUCTURE THAT MAY NOT BE IMMEDIATELY OBVIOUS TO THE APPLICATION. THE SO-CALLED "PROGRAMMER'S TRICKS" ARE OFTEN ELEGANT SOLUTIONS TO A PROGRAMMING PROBLEM; BUT BEWARE, SOME ARE ATTEMPTS BY PROGRAMMERS TO CONCEAL THEIR PROGRAMMING STATEGY.

THESE THEN, ARE GENERAL GUIDELINES TO TRY AND ATTAIN IN YOUR PROGRAMMING AND BY WHICH TO JUDGE A PARTICULAR PROGRAMMING EFFORT AS GOOD, MEDIOCRE OR POOR. NOTICE THAT THEY ARE NOT LANGUAGE SPECIFIC COMMENTS AND ARE EQUALLY APPLICABLE TO ANY PROGRAMMING LANGUAGE OR EXERCISE.

TO RETURN TO THE SIX STEPS IN THE PROGRAMMING TASK - I WILL BRIEFLY DISCUSS EACH IN TURN AND ASK THAT YOU CONSIDER EACH ONE WHEN EMBARKING UPON YOUR NEXT PROGRAMMING EXERCISE. ALSO AS ONE PROCEEDS THROUGH THE STEPS, IT IS OFTEN NECESSARY TO RECYCLE BACK THROUGH SOME OF THE PRECEEDING STEPS, TO ITERATIVELY IMPROVE THE EXERCISE AND YOUR UNDERSTANDING OF IDEAS.

1. PROJECT SELECTION

THIS MAY APPEAR TRIVIAL, BUT WE ALL HAVE TOO MANY IDEAS FOR PROGRAMS AND RARELY KNOW WHICH ONE TO TACKLE NEXT. ALSO BE HONEST WITH YOURSELF; SOME OF THE PROJECTS ARE PROBABLY TOO AMBITIOUS FOR YOUR EXISTING SKILLS AND AN ATTEMPT UPON THESE WILL POSSIBLY RESULT IN FRUSTRATION AND PERHAPS FAILURE. CHOOSE AN EXERCISE THAT IS CHALLENGING AND WORTHWHILE.

TRY NOT TO "REINVENT THE WHEEL", TRY TO BE AWARE THROUGH READING MAGAZINES OR DISCUSSING WITH OTHER USERS WHAT PROGRAMS ARE ALREADY AVAILABLE. MODIFYING AN EXISTING PROGRAM TO SUIT YOUR SPECIFICATIONS IS SOMETIMES QUICKER - IT ALSO ALLOWS YOU TO STUDY HOW OTHER PROGRAMMERS TACKLE PROBLEMS. O.K., SO NOW YOU HAVE AN IDEA OR PROBLEM THAT YOU WISH TO TACKLE AND SOLVE.

2. PROJECT FEASIBILITY

AGAIN BE HONEST. DO YOU HAVE THE HARDWARE, SOFTWARE AND KNOW-HOW TO ACHIEVE THE RESULT? ITS IS NOT REALLY MUCH USE TRYING TO WRITE LARGE BUSINESS-ORIENTED DATA BASE PROGRAMS FOR AN 8K TAPED-BASED VZ! CHECK THAT THE TASK IS REASONABLE.

3. PROJECT DEFINITION

THIS IS WHERE THE IDEA STARTS TO GET TRANSLATED INTO A REALITY. IT IS ALSO THE PHASE WHERE GENERALITY CAN BE WRITTEN IN. IT IS EASIEST TO START BY THINKING ABOUT THE INPUT TO THE PROGRAM. IS IT KEYBOARD ORIENTED, OR IS IT TO COME FROM A PROGRAMMBLE I/O PORT? PERHAPS THE PROGRAM READS ONLY DATA STATMENTS TO CONFIGURE ITSELF OR MAYBE THE PROGRAM MUST CHECK IF A PRINTER IS CONNECTED TO THE SYTEM? START DEFINING WHAT THE INPUT WILL LOOK LIKE. ASSIGN VARIABLE NAMES WITH MEANINGFUL MNEMONIC NAMES AT THIS STAGE ALSO.

NEXT, DEFINE THE OUTPUT EXPECTED FROM THE PROGRAM. IS IT TO WRITE TO TAPE AND IN WHAT FORMAT? PERHAPS IT IS TO BE SCREEN ORIENTED - CAN SOUND BE USED - OR PERHAPS VOICE SYNTHESIS TO TELL THE OPERATOR WHAT IS GOING ON? PLAN VERY CAREFULLY AND FULLY THE LAYOUT OF THE EXPECTED OUTPUT AS THIS IS HOW USERS WILL INITIALLY PERCEIVE THE QUALITY OF THE PROGRAM.

AFTER DEFINING THE I/O FOR THE PROGRAM WE SHOULD NOW HAVE A FEEL FOR THE ANTICIPATED RANGE OF PARAMETERS THAT THE PROGRAM IS MEANT TO ACCEPT AND ALSO HANDLE. THIS BRINGS IN THE VERY IMPORTANT CONCEPT OF DEFINING THE BOUNDS WITHIN WHICH THE PROGRAM MUST FUNCTION CORRECTLY. FOLLOWING ON FROM THIS, IS RANGE CHECKING OF ALL INPUT PARAMETERS SO THAT THE PROGRAM CANNOT GO BEYOND THE RANGE THAT IT WAS DESIGNED FOR AND GIVE UNEXPECTED RESULTS.

A NUMBER OF WARNING MESSAGES MUST BE BUILT INTO THE PROGRAM ALONG WITH ERROR CAPTURE AND RECOVERY ROUTINES. IT IS FAILURE TO DEFINE THE OPERATING BOUNDS OF A PROGRAM THAT CAUSES MOST CRASHES OR ROGUE BEHAVIOUR. EVEN THE DEFINITION OF INTEGER VARIABLES AT THIS STAGE CAN ASSIST BY IMPROVING PROGRAM EXECUTION TIME AND REDUCING STORAGE REQUIREMENTS.

THE DEFINITION STAGE SHOULD BE ROUGHED OUT ON PIECES OF PAPER KEPT FOR LATER REFERENCE. PERHAPS BETTER, IS TO USE AN OLD EXERCISE BOOK. ANOTHER BENEFIT OF THIS IS THAT OVER A PERIOD OF MONTHS YOUR PROGRESS CAN BE MEASURED AND YOUR GROWTH OF PROGRAMMING IDEAS RECORDED. ANOTHER BENEFIT (ALTHOUGH I HARDLY DARE MENTION IT!) IS THAT IF, AFTER THE CODING STAGE, A SYSTEM CRASH OCCURS AND YOU DIDN'T SAVE THE PROGRAM, THEN ALL IS NOT LOST - AT LEAST AN OUTLINE OF THE PROGRAM REMAINS.

4. DESIGN PHASE

HAVING SORTED OUT I/O AND OPERATING BOUNDS, THE ACTUAL SELECTION OF AN ALGORITHM TO ACHIEVE THE RESULT IS COMMENCED. BY THIS TIME SOME IDEA OF THE NUMBER OF VARIABLES REQUIRED AND THEIR TYPE SHOULD HAVE BEGUN TO GEL.

THIS IS ALSO THE STAGE WHERE YOUR BASIC HONESTY IN STAGES 1 AND 2 MAY CATCH UP WITH YOU! DATA STRUCTURE ORGANISATION AND ALGORITHM SELECTION ARE REALLY EXPERIENCE-RELATED SKILLS - HENCE THE SUGGESTION TO READ AND/OR MODIFY EXISTING PROGRAMS. BUT DO NOT DESPAIR - PRACTICE MAKES PERFECT.

5. IMPLEMENTATION PHASE

TO DATE VERY LITTLE ACTUAL CODING SHOULD HAVE BEEN DONE; IN FACT THE COMPUTER NEED NOT EVEN HAVE BEEN TURNED ON! SOME PEOPLE MAY BE SURPRISED AT HOW LATE IN THE TASK THE COMPUTER ACTUALLY ENTERS INTO THE PICTURE. AN AWFUL LOT OF PLANNING AND ORGANIZING CAN BE DONE OFF THE COMPUTER AND ON THE "BACKS OF OLD ENVELOPES".

IT IS ALSO AT THIS STAGE THAT THE CHOICE OF PROGRAMMING LANGUAGE SHOULD BE MADE. IS THE PROGRAM TIME DEPENDANT? IF IT IS, THEN IT SHOULD PROBABLY BE WRITTEN IS ASSEMBLER.

IF THE ACTUAL TIMING IS NOT SO CRITICAL THEN WRITING IN BASIC WITH ITS DIAGNOSTICS AND HELPFUL FEATURES (SO TYPICAL OF A HIGH LEVEL LANGUAGE) DEEM IT SENSIBLE. EXPERIENCED PROGRAMMERS WILL PROBABLY USE A BIT OF EACH IN PRACTICE. A VERY SENSIBLE COMPROMISE IS TO DEVELOPE THE PROGRAM IN INTERPRETED BASIC AND ONCE FINALIZED AND DEBUGGED, COMPILE THE BASIC CODE TO SPEED UP EXECUTION.

6. EVALUATION PHASE

THIS IS THE MOMENT OF TRUTH! DOES THE PROGRAM FULFIL ALL THE CRITERIA SET OUT IN THE DEFINITION PHASE? IF SO, THEN YOU HAVE SUCCESFULLY ACHIEVED YOUR TASK. IS THE OUTPUT AS YOU EXPECTED IT? ARE THE RESULTS CORRECT? IT IS A GOOD IDEA TO HAVE A STANDARD SET OF DATA TO EXERCISE THE PROGRAM SO THAT IT CAN BE QUICKLY VERIFIED AFTER A PROGRAM ALTERATION.

ENSURE THAT ALL LOGICAL PATHS THROUGH THE PROGRAM HAVE BEEN EXERCISED SO THAT NO SPURIOUS ERRORS OF LOGIC REMAIN UNDETECTED. FINALLY, DELIBERATELY TRY VALUES THAT ARE OUT OF THE INTENDED BOUNDS OF THE PROGRAM TO ENSURE THAT YOU HAVE TRAPPED THEM AND THAT THE PROGRAM RECOVERS FROM THIS TYPE OF MISUSE ABOVE AND BEYOND ITS' INTENDED DESIGN RANGE.

WP FILE RETRIEVE BY DAVE MITCHELL

EDITOR'S NOTES:

AS DAVE DIDN'T SUPPLY AN ARTICLE WITH THIS DISK UTILITY I'LL DO THE BEST I CAN WITH SUPPLIED INFORMATION. IF YOU ARE LIKE MOST DISK DRIVE USERS THEN SOONER OR LATER YOU'LL END UP WITH A CORRUPTED DISK INFORMING YOU OF THE DREADED DISK I/O ERROR. USUALLY IT'S ONLY TRACK ZERO WITH DATA TRACKS 1 TO 39 INTACT.

DAVE WIPED OUT ONE OF HIS WORD PROCESSOR FILES DISK AND THAT IS THE REASON FOR THIS WP FILE RETRIEVE ROUTINE FOR GETTING E&F 'W' WORD PROCESSOR FILES FROM DISK WHEN TRACK ZERO IS UNREADABLE. ENTER THE SOURCE CODE AND SAVE THE OBJECT CODE AS WPRET USING YOUR FAVOURITE EDITOR ASSEMBLER.

NOTE: WPRET CAN ALSO BE USED WITH R. HARRISON WORDPRO 'W' WP FILES.

USING WPRET IS VERY SIMPLE, BUT BEFORE YOU CAN USE IT TO RESURECT YOUR CORRUPTED WP FILES DISK YOU'LL NEED THE FOLLOWING INFORMATION ABOUT YOUR WP FILES.

1) FILENAME(S) ON DISK
2) START TRACK NO OF WP FILE
3) START SECTOR NO OF WP FILE

USING WPRET TO RECOVER WP FILES

1) FORMAT TRACK ZERO - (REFER TO ISSUE # 35)
2) BRUN CHECK DISK 1 - (REFER TO ISSUE # 24) OR
   BRUN CHECK DISK 2 - (REFER TO ISSUE # 42)
3) BRUN WPRET AND ENTER THE FOLLOWING INFORMATION WHEN REQUESTED:

```
ENTER FILENAME        :
ENTER START TRACK  NO:
ENTER START SECTOR NO:
```

KEEP GOING TILL THE LAST FILENAME AND CORRESPONDING DATA IS ENTERED. TO EXIT SIMPLY PRESS RETURN WHEN ASKED FOR FILENAME WITHOUT ENTERING ANYTHING. AFTER THE INFORMATION IS ENTERED ON EACH FILE WPRET WILL READ THE DISK WORKING OUT SIZE OF THE WP FILE AND THEN WRITE THE DIRectory INFORMATION AND UPDATE THE TRACK MAP.

WPRET CONTINUED PAGE 13 . . .

WP FILE RETRIEVE SOURCE CODE


```
;ROUTINE FOR GETTING
;E&F  WORDPROCESSOR
;FILES FROM DISK WHEN
;TRACK ZERO IS UNREADABLE
        LD    HL,A0
        LD    (788EH),HL
A0      LD    HL,END
        LD    (ADD),HL
        LD    HL,M1
        CALL  IPUT
        CALL  ATB
        LD    (ADD1),A
        LD    HL,M2
        CALL  IPUT
        CALL  ATB
        LD    (ADD2),A
        LD    HL,M4
        CALL  IPUT
        LD    A,20H
        LD    (DE),A
        PUSH  IY
        POP   DE
        INC   DE
        LD    BC,8
        LDIR
        DI
        CALL  4008H
        LD    BC,0032H
        CALL  4038H
        CALL  4011H
        OR    A
        JP    NZ,400EH
        LD    A,(ADD1)
        LD    (IY+12H),A
        LD    A,(ADD2)
        LD    (IY+11H),A
A1      LD    L,(IY+34H)
        LD    H,(IY+35H)
        LD    A,(IY+12H)
        DEC   A
        SLA   A
        LD    E,A
        LD    D,0
        LD    A,(IY+11H)
        CP    8
        CCF
        ADC   HL,DE
        AND   7
        INC   A
        LD    B,A
        LD    C,(HL)
        RLC   C
A13     RRC   C
        DJNZ  A13
        SET   0,C
        LD    B,A
        RRC   C
A14     RLC   C
        DJNZ  A14
        LD    (HL),C
        CALL  4035H
        OR    A
        JP    NZ,400EH
        LD    L,(IY+31H)
        LD    H,(IY+32H)
        LD    DE,(ADD)
        LD    BC,007EH
        LDIR
        LD    (ADD),DE
        LD    A,(HL)
        OR    A
        JR    Z,A2
        LD    (IY+12H),A
        INC   HL
        LD    A,(HL)
        LD    (IY+11H),A
        JR    A1
A2      DEC   DE
        LD    A,(DE)
        OR    A
        JR    Z,A2
        INC   DE
        LD    HL,END
        EX    DE,HL
        XOR   A
        SBC   HL,DE
        EX    DE,HL
        LD    HL,0D000H
        XOR   A
        SBC   HL,DE
        LD    (ADD),HL
        CALL  402CH
        OR    A
        JP    NZ,400EH
        CALL  402FH
        OR    A
        JP    NZ,400EH
        LD    (HL),57H
        INC   HL
        LD    (HL),3AH
        INC   HL
        PUSH  HL
        POP   DE
        PUSH  IY
        POP   HL
        INC   HL
        LD    BC,8
        LDIR
        LD    A,(ADD1)
        LD    (DE),A
        INC   DE
        LD    A,(ADD2)
        LD    (DE),A
        INC   DE
        LD    HL,(ADD)
        LD    A,L
        LD    (DE),A
        INC   DE
        LD    A,H
        LD    (DE),A
        INC   DE
        LD    A,0
        LD    (DE),A
        INC   DE
```

43-13 WPRET SOURCE CODE CONTINUED

```
        LD    A,0D0H
        LD    (DE),A
        CALL  4032H
        OR    A
        JP    NZ,400EH
        CALL  4017H
        OR    A
        JP    NZ,400EH
        CALL  400BH
        CALL  3450H
        EI
        JP    A0
IPUT    CALL  2B75H
        LD    A,(053AH)
        CP    3AH
        JR    Z,B0
        CALL  0545H
        JR    B1
B0      CALL  053AH
B1      JP    C,1DBEH
        LD    A,(DE)
        CP    20H
        JP    Z,1A19H
        INC   HL
        RET
ATB     LD    B,2
        LD    DE,0
ATB1    LD    A,(HL)
        CP    30H
        JR    C,ATB4
        CP    3AH
        JR    C,ATB2
        CP    41H
        JR    C,ATB4
        CP    47H
        JR    NC,ATB4
        SUB   7
ATB2    AND   0FH
        LD    C,B
        LD    B,4
ATB3    SLA   E
        RL    D
        DJNZ  ATB3
        OR    E
        LD    E,A
        LD    B,C
        RST   10H
        DJNZ  ATB1
ATB4    LD    A,B
        CP    2
        JP    NC,1997H
        LD    A,E
        RET
ADD     DEFW  0
ADD1    NOP
ADD2    NOP
M1      DEFB  1FH
*RETRIEVER FOR E&F WORDPROC*
*ESSOR*
        DEFB  0DH
*FILES ONLY     (D. MITCHELL)
        DEFW  0D0DH
* ENTER START TRACK    :*
        NOP
M2      EQU   $
* ENTER START SECTOR   :*
        NOP
M4      EQU   $
* ENTER FILENAME :*
        NOP
END     NOP
```

WP FILE RETRIEVE CONTINUED

WHENEVER POSSIBLE I LIKE TO TEST PROGRAMS BEFORE PUBLICATION. AS I DID NOT HAVE A CORRUPTED WP FILE DISK I TRIED IT ON ONE MY JOURNAL WP FILE DISKS BY FORMATTING TRACK ZERO, RUNNING CHKDSK AND THEN WPRET. I AM PLEASED TO ANOUNCE THAT WPRET RESTORED THE DIRECTORY TRACK AND ALL WP FILES WERE 100% OK.

IT IS ESSENTIAL FOR THE RECOVERY PROCESS TO HAVE THE FILENAME(S), START TRACK & SECTOR NUMBER OF YOUR WP FILES WHICH CAN BE OBTAINED BY USING ONE OF THE FOLLOWING UTILITIES:

A) DAVE MITCHELL EXTENDED DOS VERSION 1.3 OR
   " " EXTENDED DOS V1.4 - 6000H VERSION

B) " " DISK FILER CATALOGUER
   AVAILABLE FROM DAVE MITCHELL

C) LESLIE MILBURN EXTENDED DOS VERSION 12.2
   AVAILABLE FROM THE EDITOR

D) LARRY TAYLOR FILE SEARCH VERSION 2.0
   AVAILABLE FROM THE EDITOR

ALL THE UTILITIES MENTIONED WILL PROVIDE THE INFORMATION REQUIRED SIMILIAR TO WHAT IS SHOWN BELOW. IF YOU'RE LIKE ME THEN YOU'LL HAVE A MIXTURE OF FILES ON THE ONE DISK. I TRIED WPRET PROCEDURE AND FOUND THAT WPRET IS MORE VERSATILE THAN I FIRST THOUGHT.

BEFORE WPRET PROCEDURE

```
T:PLOTTO-2 01 00 7AE9 98D7 1DEE
B:PARK2    04 0D 9000 90E3 00E3
B:DIRA     04 0F 7200 72E5 00E5
D:LOTTO    05 01 0000 0000 0000
B:MAPAUST  05 04 7000 7800 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

AFTER WPRET PROCEDURE

```
W:PLOTTO-2 01 00 B212 D000 1DEE
W:PARK2    04 0D CF1D D000 00E3
W:DIRA     04 0F CF1B D000 00E5
W:LOTTO    05 01 CF00 D000 0100
W:MAPAUST  05 04 C800 D000 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

AFTER FILETYPE CHANGE

```
T:PLOTTO-2 01 00 B212 D000 1DEE
B:PARK2    04 0D CF1B D000 00E3
B:DIRA     04 0F CF1B D000 00E5
D:LOTTO    05 01 CF00 D000 0100
B:MAPAUST  05 04 C800 D000 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

AFTER START ADDRESS CHANGE

```
T:PLOTTO-2 01 00 7AE9 98D7 1DEE
B:PARK2    04 0D 9000 90E3 00E3
B:DIRA     04 0F 7200 72E5 00E5
D:LOTTO    05 01 0000 0100 0100
B:MAPAUST  05 04 7000 7800 0800
W:INST-1   06 05 CB52 D000 04AE
W:INST-2   06 0F C927 D000 06D9
7 FILE(S)        64.375K FREE
```

BEFORE WPRET PROCEDURE

THIS SHOWS DIRECTORY PRINTOUT OF DISK BEFORE EXPERIMENTATION WITH ALL INFORMATION NEEDED TO RESURECT ALL FILES.

AFTER WPRET PROCEDURE

I FORMATTTED TRACK ZERO, RAN CHKDSK AND THEN WPRET AND PRINTOUT SHOWS RESULT. PLEASE NOTE THAT ALL FILES HAVE 'W' FILETYPE AND END ADDRESS OF D000 WHICH IS STANDARD FOR DAVE MITCHELL WP FILES. ALSO NOTE THAT WPRET WORKED OUT SIZE OF FILES WHICH IS SHOWN ALSO.

AFTER FILETYPE CHANGE

THE TWO WP FILES INST-1 & INST-2 NEED NO CHANGING AT ALL TO BE USED. THE REST DO THOUGH AND I USED LESLIE MILBURN'S EXT12.2 TO CHANGE FILETYPES AND ONCE AGAIN PRINTOUT SHOWS RESULT. PLEASE NOTE SYNTAX FOR FILETYPE CHANGE:

```
CHA"PLOTTO-2",T
```

AFTER START ADDRESS CHANGE

AFTER CHANGING FILETYPES I CHANGED START ADDRESSES USING THE FOLLOWING SYNTAX:

```
REL"PLOTTO-2",7AE9
```

THE LAST DIRECTORY PRINTOUT SHOWS RESULT. IF YOU COMPARE IT WITH THE FIRST YOU'LL NOTE IT IS IDENTICAL EXCEPT FOR THE DATA FILE WHICH HAS AN END ADDRESS OF 0100 WITH CORESPONDING FILE SIZE. IT IS NO PROBLEM AS IT LOADED OK WITH NO CORRUPTION EVIDENT IN IT AND ALL OTHER FILES AS WELL.

43-15 WP FILE RETRIEVE CONTINUED

IT TOOK A BIT OF FIDDLING TO RESURECT TRACK ZERO, BUT IT WAS SUCCESFUL. I COULD HAVE USED FORMAT FROM ISSUE # 35 AND LABEL2 FROM ISSUE # 37 WHICH WOULD HAVE BEEN EASIER AND QUICKER AS I HAD ALL THE RELEVANT FILE INFORMATION. THE REASON I CHOSE WPRET IS TO SHOW ITS VERSATILITY.

IN NEXT ISSUE I'LL SHOW YOU HOW TO USE WPRET PROCEDURE TO RESURECT DISKS WHOSE TRACK ZERO IS CORRUPTED AND YOU DO NOT HAVE MINIMUM DIRECTORY INFORMATION LIKE FILENAME & START TRACK/SECTOR NO'S. ALSO THERE'S A VERY SIMPLE WAY YOU COULD TRY FIRST WHICH WORKS QUITE WELL. MORE ON IT NEXT ISSUE.

SUMMARY: WPRET WAS DESIGNED TO RESURECT 'W' FILETYPE WP FILES ON A DISK WHOSE TRACK ZERO IS CORRUPTED. WPRET COULD BE MODIFIED FOR OTHER SPECIAL DISKS OF THE SAME FILETYPE OR USED TO UPGRADE RETRIEVE FROM ISSUE # 25.

PREVENTITIVE DISK MAINTAINANCE

THERE IS A LOT YOU CAN DO TO SAFEGUARD YOUR DISKS AND THE PRECIOUS DATA ON THEM AND TRY TO PREVENT UNINTENTIONAL OR ACCIDENTAL DISK CORRUPTION. YOU CAN DO SOME OF THE FOLLOWING TO MINIMISE THAT POSSIBILITY AND TO PROVIDE INFORMATION FOR RETRIEVAL IN CASE THE WORST HAPPENS AND YOU END UP WITH A CORRUPTED DISK.

1) MAKE DIRECTORY PRINTOUTS OF YOUR DISKS SHOWING AT MINIMUM FILETYPE, FILENAME, START TRACK AND SECTOR NO'S.

   THE MOST SUITABLE FOR THE PURPOSE IS DAVE MITCHELL'S DISK FILER & CATALOGUER AS IT CAN PRINTOUT ALL YOUR DISKS FROM A DATA FILE OR JUST A SINGLE DISK. PREVIOSLY MENTIONED UTILITIES ON PAGE 13 ARE ALSO SUITABLE AS THEY ALL GIVE THE REQUIRED INFORMATION FOR POSSIBLE FUTURE RETRIEVAL.

2) BACK UP YOUR DISKS TO FLIP SIDE BY NOTCHING OTHER SIDE OF DISK AND DCOPY DISK TO OTHER SIDE OR USE ONE OF THE DISK COPYING PROGRAMS OF WHICH THERE ARE A FEW AROUND.

3) BACK UP YOUR WORK AT REGULAR INTERVALS BE IT A WP OR DATA FILE OR THE PROGRAM YOU ARE WORKING ON AS IT IS EASIER THAN LOSING IT ALL AND DOING IT ALL OVER AGAIN. SAVES A LOT OF EXTRA WORK.

4) USE WRITE PROTECT LABELS TO SAFEGUARD YOUR VALUABLE PROGRAM AND DATA DISKS AGAINST ACCIDENTAL CORRUPTION.

5) CLOSE DRIVE DOOR FOR DISK ACCESS ONLY.
   LEAVE DRIVE DOOR OPEN AT ALL OTHER TIMES.
   FAILURE TO DO THIS RISKS WIPING OUT TRACK ZERO OR WHOLE DISK.

6) KEEP YOUR DISKS AWAY FROM HARMFUL ENVIROMENTS OF WHICH SOME EXAMPLES ARE SHOWN BELOW:

A) KEEP AWAY FROM MAGNETIC FIELDS AND METAL OBJECTS.
B) KEEP AWAY FROM SEVERE HEAT OR COLD.
C) STORE DISKS IN THEIR JACKETS AND IN UPRIGHT POSITION.
D) NEVER TOUCH DISK SURFACE WITH YOUR FINGERS OR ANY OBJECT.
E) DO NOT SMOKE IN THE VICINITY OF DISKS OR COMPUTERS.
F) DO NOT USE BIRO OR OTHER HARD WRITING IMPLEMENT TO WRITE ON DISK LABEL AS IT COULD POSSIBLY DAMAGE DISK SURFACE. USE FELT PEN.

CONTRIBUTIONS TO THE JOURNAL

IF YOU ARE THINKING OF CONTRIBUTING TO THE JOURNAL THE PREFERED FORMAT IS BASIC LISTINGS, WORD PROCESSOR OR SOURCE CODE FILES ON TAPE OR DISK. FILES FROM THE FOLLOWING WORD PROCESSORS CAN BE ACCEPTED :-

E & F TAPE OR DISK PATCH 3.1-3.3, WORDPRO CARTRIDGE, WORDPRO PATCH, ALL SOURCE CODE FILES AND ALL QUICKWRITE WORD PROCESSOR FILES.

CLUB MEETINGS - ALL WELCOME

MEETINGS WILL BE APPROXIMATELY ONCE A MONTH. BECAUSE SOME LOCAL MEMBERS WORK SHIFTWORK MEETING DATES WILL BE ADJUSTED TO ACCOMODATE THEM. WHETHER YOU ARE A LOCAL MEMBER, INTRA OR INTERSTATE VISITOR PLEASE CHECK WITH JOE LEON FIRST.

HUNTER VALLEY VZ JOURNAL SUBSCRIPTIONS

SUBSCRIPTION TO - AUST. - 3 ISSUES $11.00 - 6 ISSUES $21.00
H.V.VZ.JOURNAL - N. Z. - 3 ISSUES $13.00 - 6 ISSUES $26.00
PRICES INCLUDE POST & PACKING

FOR MORE INFORMATION ON MEETINGS AND SUBS CONTACT:
JOE LEON 33 TIGHES TCE TIGHES HILL 2297 (049) 692 399 AUSTRALIA

VZ USER GROUPS & PUBLICATIONS

VZ DOWN UNDER & VZ DISK MAGAZINE HAVE CEASED PRODUCTION

WAVZ - WESTERN AUSTRALIA VZ USER GROUP
GRAEME BYWATER PO BOX 388 MORLEY W A 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178
SOFTWARE FOR SALE - DISK MENU

NOTE: WHEN WRITING TO ANY ABOVE OR H.V.VZ. USERS' GROUP FOR INFORMATION PLEASE ENCLOSE A S.S.A.E. OR NZ 2 INT. REPLY COUPONS.

* INDEX *
JUNE 86 - JUNE 93

HUNTER VALLEY
VZ USERS GROUP
NEWSLETTER

ISSUES 1-4

HUNTER VALLEY
VZ USERS GROUP
MAGAZINE

ISSUES 5-6

HUNTER VALLEY
VZ JOURNAL

ISSUES 7-44

HUNTER VALLEY VZ JOURNAL INDEX 43-01

43-02 HUNTER VALLEY VZ JOURNAL INDEX

ISSUE-PAGE CONTENTS

HUNTER VALLEY VZ JOURNAL INDEX

HUNTER VALLEY VZ JOURNAL INDEX 43-05

CONTINUED NEXT ISSUE